Data Structures and Algorithms (DT133G/DT127G)

First Examination VT2020 (Exam #46)

Teacher:	Federico Pecora (ext. 3319)
Published:	2020-03-24 at 08:15
No. exercises:	6
Total points:	70 (42 required to pass)
No. pages:	6 (excluding this page)

- This is a **personalized exam**
- Compile your answers into a **Word** or **PDF** document
- Submission instructions
  - Open the course Blackboard page, go to the Course material section
  - Click on the item entitled **Exam 2020-03-24 assignment**
  - Submit your Word/PDF file as an **attachment** to this assignment
- Submissions are due **at 12:15 today** (unless you are given another deadline by Funka)
- The teacher is available **from 9:00 to 10:30 to answer questions** at:
  - Zoom (browser): `https://oru-se.zoom.us/j/948414044`
  - Telephone: `+46850520017` or `+46850539728` (meeting ID: 948414044)
  - Mobile: `+46850520017,,948414044#` or `+46850539728,,948414044#`
  - If you are put on hold, please wait, as this means there is a queue
- If further doubts arise, make reasonable assumptions and **write them down**

Exercise 1 (10 points)

Given the array $A = \langle 14, 16, 5, 7, 9, 11, 8, 15, 4, 6 \rangle$, show how the Mergesort and Insertion Sort algorithms work. Step through each algorithm, illustrating how it modifies the input array A. State the worst- and best-case computational complexity of the two algorithms in terms of the size $|A|$ of the input array, and explain why.

Exercise 2 (12 points)

Construct a Binary Search Tree (BST) from the array of integers A in the previous exercise. Do this by inserting each integer into the tree, one by one, starting from the left (element '14'). Draw the resulting tree and state its height. Is the tree balanced?

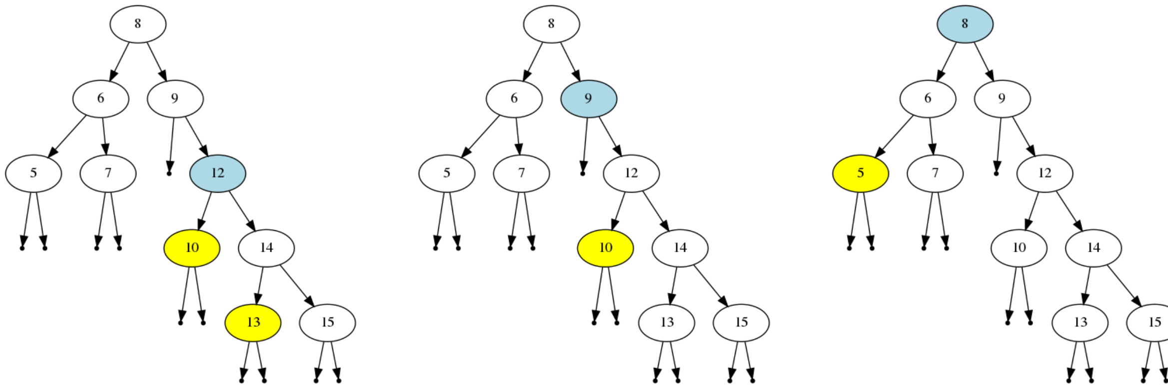
Remove the third element of A (element '5') from the BST, explaining the operations that this procedure performs. Now, insert the element you removed back into the BST, and draw the resulting tree. Is the resulting BST different from the BST before the element was removed? If so, will this be the case for any element that is removed and then re-inserted?

Illustrate the algorithm for finding the successor of an element in a BST. Show how this works, step by step, for the fourth element of A (element '7'). In general, how many key comparisons are performed by the algorithm for finding the successor of an element in a BST? What is the computational complexity of the algorithm?

Construct a new BST using the sorted array you obtained in the Exercise 1, again by inserting the elements of the array into the tree one by one. What is the heght of this BST, and is it balanced?

Exercise 3 (12 points)

The Deepest Common Ancestor (DCA) of two nodes x and y in a Binary Search Tree (BST) is the node that is the common ancestor for both x and y, and is the farthest away from the root node of the BST. Note that one of the two given nodes x and y could itself be the DCA of the other node, or the DCA could be a third node z. The three possible cases are shown in Figure 1.



(a) (b) (c)

Figure 1: (a) 12 is the DCA of 10 and 13; (b) 9 is the DCA of 9 and 10; (c) 8 is the DCA of 5 and 8.

Your task is to design an algorithm for finding the DCA z of a given pair of nodes x and y in a given BST. Consider the following facts:

- Assume that $x < y$ and that z is the root of the BST.
- If x and y are both less than z, then the DCA of x and y is in the left sub-tree of z.
- If x and y are both major than z, then the DCA of x and y is in the right sub-tree of z.
- If x is less than z and y is major than z, then z is the DCA of x and y.

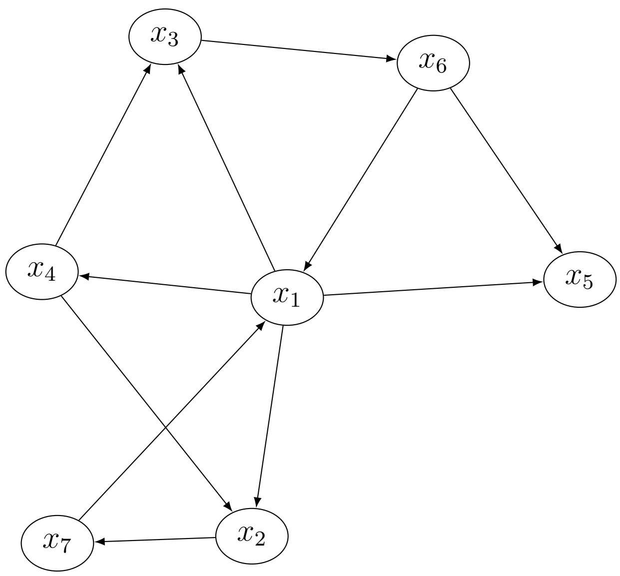
Write the pseudocode of the algorithm, and show its worst-case computational complexity. Illustrate how the algorithm works, step by step, on the first BST you obtained in the previous exercise (the one from the original unsorted array A of Exercise 1). Choose your inputs to be $x = 6$ and $y = 16$.

Exercise 4 (12 points)

Given a directed, unweighted graph $G = (V, E)$, describe an algorithm to solve each of the following problems:

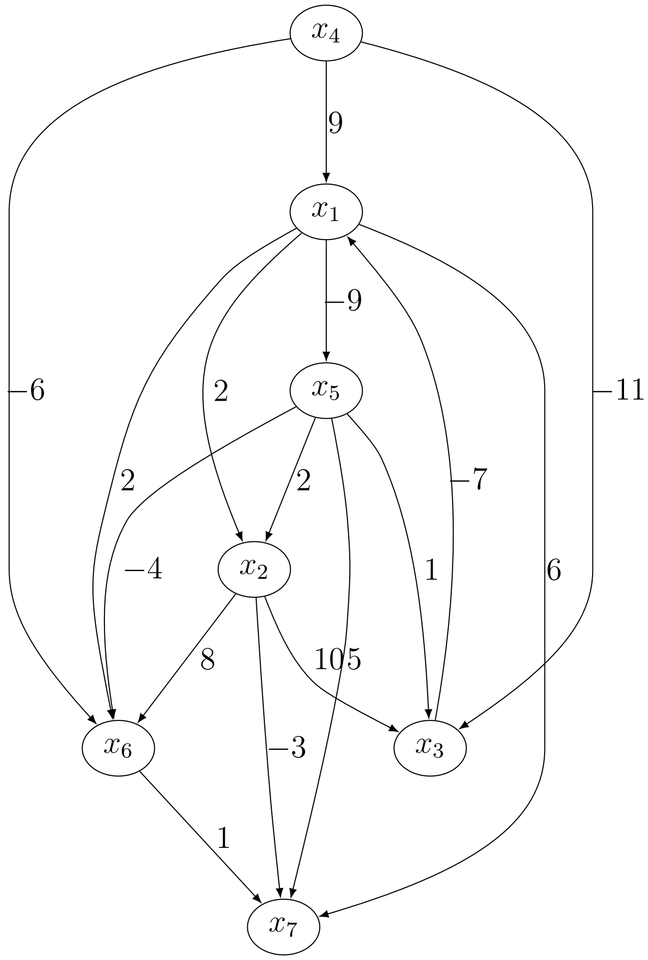
- Return whether the graph is connected.
- Return whether the graph is a tree.
- Find all the strongly connected components of the graph.
- Find the shortest path from node x_1 to every other node.

Show all the steps performed by each algorithm, using as input the graph G given below. Also, state and explain the computational complexity required to solve each problem in terms of the number of nodes $|V|$ and number of edges $|E|$ of the input graph.



Exercise 5 (12 points)

Given the following directed graph $G = (V, E)$ with weight funtion $w : E \mapsto \mathbb{Z} \setminus \{0\}$ as specified on the edges of the graph, illustrate an algorithm for finding a shortest path from vertex x_1 to every other vertex in the graph. Show how, at each step, the algorithm updates the distance estimate $d[x_i]$ of every vertex $x_i \in V$. State the computational complexity of the algorithm in terms of the number of nodes $|V|$ and number of edges $|E|$ of the input graph.



Exercise 6 (12 points)

State whether each of the following 6 statements on the asymptotic behavior of the function f is true or false.

1.	$f(n) = 9^n$,	$g(n) = 6^n$,	$f(n) \in \omega(g(n))$
2.	$f(n) = 4n^3 - 4n^2 - 5n$,	$g(n) = 4^n$,	$f(n) \in \Theta(g(n))$
3.	$f(n) = 7\log(n)$,	$g(n) = 8^n$,	$f(n) \in \omega(g(n))$
4.	$f(n) = 6^n$,	$g(n) = 8\log(n)$,	$f(n) \in \omega(g(n))$
5.	$f(n) = 3\log(n)$,	$g(n) = 2\log(n)$,	$f(n) \in \omega(g(n))$
6.	$f(n) = 2n\log(n)$,	$g(n) = 7\log(n)$,	$f(n) \in o(g(n))$